

REMARKS/ARGUMENTS

I. Summary of Claim Status

Claims 1-32 remain in this application as originally filed.

In the office action dated January 6, 2004, the Examiner rejected all 32 pending claims. Claims 1-6, 11-13, 17-19 and 21-25 were rejected under 35 U.S.C. 102(e) as being anticipated by Schloss et al., U.S. Patent No. 6,249,844 ('Schloss'). Claims 7-10, 14-16, 20 and 26-32 were rejected under 35 U.S.C. 103(a) as being unpatentable over Schloss. The applicant acknowledges with appreciation, the Examiner's detailed description of the manner in which the references were applied to the claims. However, the cited references do not appear to anticipate or render any of the claims obvious.

II. The Disclosure of Schloss

Schloss addresses the need of creating one or more persistent object fragments from a named SGML, HTML or XML document or object which might include an object containing dynamic information, such as price data from a database, to facilitate caching of the object fragment that contains the non-dynamic (persistent) information contained in the object. *Schloss* 2:8-30, 53-61. *Schloss* further states that:

FIG. 3 depicts an example of a **document** with 3 SEGMENTS. As discussed, the first segment (330₁) begins with a start-tag, <cml: molecule>, and ends with an end-tag, </cml: molecule> and the second segment begins with a start-tag, <m: order>, and ends with an end-tag, </m: order>. The second segment (330₂) includes a third segment (330₃) nested within it. The third segment begins with a start-tag, <db: price>, and finishes with an end-tag, </db: price>. Assume the semantics of the three segments as follows. Assume the first segment provides an image of a molecule structure of a chemical compound. Assume also the second segment contains a formula to generate an order table showing the price at

different quantities. **Assume further, the third segment retrieves the price information from the product database. Hence it is a segment with dynamic information.**

FIG. 4 depicts an example of a **modified Web document** after the persistent fragments have been recognized and extracted. Here it is assumed that generating the molecular structure of the chemical compound in the first segment (330₁) is quite complex, whereas the computation of the order table is straightforward. Hence, only the first (330₁) and the third segments (330₃) are recognized as persistent fragments with the identities, "125.1" and "28.3", respectively. In the preferred embodiment, **each of the persistent fragments is replaced with an "include" statement referring to the name of the fragment, e.g. <include HREF="125.1">, indicating the reference to the fragment "125.1," and followed by a <include> statement.** *Schloss* 5:37-52 (emphasis added)

Those of ordinary skill in the art to which the invention relates will recognize that a markup language document or object is not a database. *Schloss* does not parse the data in the database which includes static and dynamic data but rather parses a document or object which refers to dynamic data in the database to eliminate the dynamic data from the new persistent fragment of the document or object and replace it with an "include" statement referring to the name of the fragment. Thus, although a computer aided search of the patent database might establish that *Schloss* contains the words or phrases "database," "dynamic data" and "static data," *Schloss* can in no way be interpreted as disclosing or even suggesting each element and limitation of any of the independent claims of the present application.

III. Schloss does not anticipate claims 1-6, 11-13, 17-19 or 21-25

It is well settled law that a prima facie case of anticipation of a claim by a reference requires that the reference disclose, either literally or inherently, every element and limitation of the claim. For a limitation to be inherently disclosed:

it must be necessarily present and a person of ordinary skill in the art would recognize its presence. *In re Robertson*, 169 F. 3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999); *Continental Can*, 948 F.2d at 1268, 20 USPQ2d at 1749. Inherency "may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. *Id.* at 1269, 20 USPQ2d at 1749 (quoting *In re Oelrich*, 666 F.2d 578, 581, 212 USPQ 323, 326 (CCPA 1981). *Crown Operations International Ltd. v. Solutia Inc.*, 62 USPQ2d 1917, 1922 (Fed. Cir. 2002).

A. Schloss does not anticipate independent claims 1, 12 or 23 or dependent claims 2-6 and 11, which depend from claim 1, dependent claims 13, 17-19 and 21-22, which depend from claim 12, or dependent claim 24, which depends from claim 23.

Independent claims 1, 12 and 23 recite several common elements and limitations not disclosed by Schloss. These independent claims recite:

1. A data storage system comprising: a **database partitioned into a first section and a second section, said first section comprising static data and being stored in a static memory device, said second section comprising dynamic data and being stored in a dynamic memory device**; and, a database manager for managing said database.

12. A control system having a data storage system for storing data related to said control system, the control system comprising: a communication network; an application node coupled to said communication network, said application node having a static memory device and a dynamic memory device; a **database partitioned into a first section and a second section, said first section comprising static data and being stored in said static memory device, said second section comprising dynamic data and being stored in said dynamic memory device**; and, a database manager disposed in said application node for managing said database.

23. A method for creating a **database**, said method comprising the steps of: storing a set of **static data elements in a static memory device**; and, storing a set of **dynamic data elements in a dynamic memory device**, wherein said database comprises said static data elements and said dynamic data elements.

Thus, independent claims 1, 12 and 23 each recite a database comprising static data stored in the static memory device and the dynamic data stored in the dynamic

memory device. *Schloss* does not disclose such a partitioned database, but rather discloses an original SGML, HTML or XML document or object including references to objects that may reference dynamic data stored in a field of a database and objects that may reference static data stored elsewhere, and an object fragment including only certain static objects of the original SGML, HTML or XML and "include" statements referencing dynamic data. *Schloss* does not disclose a partitioned database or even modify the database mentioned therein. The data in the database in *Schloss* remains stored in the type of memory device in which it was originally stored.

Additionally, independent claims 1, 13 and 23 require the presence of a set of static data stored in a static memory device and set of dynamic data stored in a dynamic memory device. While the language cited by the examiner in *Schloss* does indicate that dynamic and static data might be distinguished, there is no indication of a static memory device and a dynamic memory device, much less any indication that a set of static data is stored in the static memory device while a set of dynamic data is stored in a dynamic memory device. *Schloss* does indicate that an object fragment of an XML, SGML or HTML document may be created that eliminates the references to dynamic data objects and retains the references to certain static data objects so that the object fragment may be stored in cache. As an example, *Schloss* states that "the object description can include a dynamic part which would otherwise prevent the object from being cached. The dynamic part can be recognized and treated as a separate fragment from the object description. Thus the revised document becomes static and therefore cacheable." *Schloss*, Abstract. This does not necessarily result in static data being stored in a static memory device and dynamic data being stored in a dynamic memory device as

recited in claims 1, 12 and 23. In fact, *Schloss* appears to envision that the actual data which is referenced as an object in the XML, SGML or HTML document, whether static or dynamic, remain stored in whatever memory device in which it was previously stored while only an object fragment of the original XML, SGML or HTML document including the references to certain static data elements be stored in cache. Thus, the data from the database referred to in *Schloss*, which is identified as dynamic data, remains stored in the database and is not even retrieved by the object fragment for storage in the cache.

Since *Schloss* does not disclose, either literally or inherently, every element and limitation of independent claims 1, 12 and 23, *Schloss* does not anticipate those independent claims or claims 2-6, 11, 13, 17-19, 21, 22 or 24 which depend therefrom.

B. Schloss does not anticipate independent claim 25.

Claim 25 recites:

25. A method for editing a data element stored in a static memory device comprising a plurality of storage units, said method comprising the steps of: copying a content of one of said storage units to a dynamic memory device, wherein said content comprises said data element; editing said data element while said data element is stored in said dynamic memory; erasing said one of said storage units; and, writing said content, including said data element that has been edited, into said one of said storage units.

Schloss does not envision editing a data element stored in static memory in the manner recited in claim 25.

IV. Schloss does not render any of the pending claims, and in particular claims 7-10, 14-16, 20 and 26-32 obvious

In order to establish a prima facie case of obviousness . . . the prior art reference (or references when combined) must teach or suggest all the claim limitations. . . . The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." Ex parte Clapp, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). M.P.E.P. 706.02(j).

A. Schloss does not render claims 7-10, 14-16 or 20 obvious.

Claims 7-10 depend from claim 1 and claims 14-16 and 20 depend from claim 12. As mentioned above in section III.A., Schloss does not disclose every element and limitation of claims 1 or 12. The Examiner's comments with regard to each of these claims individually does not cure the lack of disclosure of the database comprising static data stored in the static memory device and the dynamic data stored in the dynamic memory device or the lack of disclosure of the presence of a set of static data stored in a static memory device and set of dynamic data stored in a dynamic memory device. If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

B. Schloss does not render independent claim 27, or dependent claims 28-31 which depend from claim 27, obvious.

Claim 27 which recites:

27. A computer program product comprising a computer readable code stored on a computer readable medium, that when executed, causes a computer to: read a catalog to determine where a set of static data shall be stored in a static memory device; **store said static data in said static memory device** according to said catalog; read said catalog to determine where a set of dynamic data shall be stored in a dynamic memory device; and **store said dynamic data in said dynamic memory device** according to said catalog.

Thus claims 27-31 require the presence of a set of static data stored in a static memory device and set of dynamic data stored in a dynamic memory device.

While the language cited by the Examiner in *Schloss* does indicate that dynamic and static data might be distinguished, there is no indication of a static memory device and a dynamic memory device, much less any indication that a set of static data is stored in the static memory device while a set of dynamic data is stored in a dynamic memory device. *Schloss* does indicate that an object fragment of an XML, SGML or HTML document may be created that eliminates the references to dynamic data objects and retains the references to certain static data objects so that the object fragment may be stored in cache. As an example, *Schloss* states that "the object description can include a dynamic part which would otherwise prevent the object from being cached. The dynamic part can be recognized and treated as a separate fragment from the object description. Thus the revised document becomes static and therefore cacheable." *Schloss*, Abstract. This does not necessarily result in static data being stored in a static memory device and dynamic data being stored in a dynamic memory device as recited in claims 27-31. In fact, *Schloss* appears to envision that the actual data which is referenced as an object in the

XML, SGML or HTML document, whether static or dynamic, remain stored in whatever memory device in which it was previously stored while only an object fragment of the original XML, SGML or HTML document including the references to certain static data elements be stored in cache. Thus, the data from the database referred to in *Schloss*, which is identified as dynamic data, remains stored in the database and is not even retrieved by the object fragment for storage in the cache.

C. Schloss does not render claims 26 or 32 obvious

Claims 26 and 32 that are directed to compressing databases by identifying Boolean elements which require less storage capacity than provided in a typical data field and collecting several Boolean elements in a single data field. Claims 26 and 32 recite:

26. A method performed by a database generation tool for creating a **compressed database**, said method comprising the steps of: receiving a data input file, said data input file defining a first set of data fields to be included in said database and said data input file including a set of data elements to be included in said database; **identifying a second set of data fields in said data input file that are designated to contain a Boolean element**, said second set of data fields being a subset of said first set of data fields; **defining one or more new data fields for collectively storing said Boolean elements**; modifying said first set of data fields to eliminate said second set of data fields; and, generating a catalog that defines an arrangement of said first set of data fields, wherein said arrangement includes said one or more new data fields for collectively storing said Boolean elements.

32. A computer program product comprising a computer readable code stored on a computer readable medium, that when executed, causes a computer to: receive a data input file that defines a first set of data fields to be included in a database, said data input file including a plurality of data elements to be included in a database; use said data input file to identify a **second set of data fields that are each designated in said data input file for storing a Boolean element**, said second set of data fields being a subset of said first set of data fields; modify said

first set of data fields to eliminate said second set of data fields; and, create a catalog for said database, said catalog defining an arrangement of said first set of data fields, wherein said arrangement includes said **one or more new data fields for collectively storing said Boolean elements**.

As mentioned above, *Schloss* addresses modification of SGML, HTML or XML documents or objects to remove references to objects that contain dynamic data so that the modified document or object is cacheable. While *Schloss* does indicate as an example that an object containing dynamic data could include a reference to a price field in a database, *Schloss* does not envision modifying or compressing the database in any manner. The price field referred to in *Schloss* would not even be a Boolean element (unless of course only two prices are ever charged for product, in which case a Boolean element could be used to represent a price). Regardless of whether the price field could be represented as a Boolean element or not, *Schloss* nowhere indicates that various Boolean elements in a set of data fields could be collected into a new data field to compress a database. In fact, *Schloss* does not envision collecting data of any specific type and storing it in a single field that would result in compression of the data.

Applicant reserves the right to assert additional bases for establishing the patentability of claims 14-16, 20 and 26-32 over the Schloss, including, but not limited to, the lack of disclosure of additional elements and limitations of the claims, whether there is a teaching, motivation, or suggestion to select and combine the reference relied on as evidence of obviousness with common knowledge.

V. Conclusion

Schloss does not teach or suggest every element and limitation recited in any of the thirty-two pending claims. Thus, claims 1-32 are in condition for allowance. Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

While applicant believes that this is a timely response filed within three months of the mailing date of the office action addressed herein and thus no fee is due for this response, authorization is hereby provided to charge any additional fees required to Deposit Account No. 13-0014, but not to include any payment of issue fees.

Respectfully submitted,
MAGINOT, MOORE & BECK

By David B. Quick
David B. Quick
Reg. No. 31,993
Tel.: (317) 638-2922